Section I:

AMENDMENT UNDER 37 CFR §1.121 to the

CLAIMS

1. (currently amended) A logical device for handling dynamic attributes in a static directory comprising:

a set of attribute declarations containing at least one declaration for [[an]] directory attribute to be handled dynamically as a ~~dynamic~~ real-time attribute whose value is retrievable outside of static memory of a directory structure;

at least one Real-time Attribute Processor (RTAP) configured to ~~determine~~ dynamically resolve a ~~dynamic~~ real-time value for an attribute declared as being ~~dynamic~~ real-time in said set of attribute declarations by obtaining an attribute value from a real-time source external to said directory structure, and by converting said obtained value to conform to a directory request return format;

an RTAP selector configured to select and invoke an RTAP according to a predetermined selection schema; and

a directory attribute processor configured to parse requests for access to directory attribute values, to detect requests for attributes declared as ~~dynamic~~ real-time in said attribute declarations, to operate said RTAP selector to invoke a corresponding RTAP, to receive an attribute value determined by said invoked RTAP, and to return said real-time attribute value to a requester.

2. (currently amended) The logical device as set forth in Claim 1 wherein said directory attribute processor is further adapted to suppress storage of said resolved attribute value in a directory.

3. (currently amended) The logical device as set forth in Claim 1 wherein said RTAP selector is ~~adapted~~ configured to select an RTAP based upon a variation of a name of said requested directory attribute.

4. (original) The logical device as set forth in Claim 3 wherein said name variation comprises a name identifying a function selected from the group of a logical device, a device address, a name of a JAVA class, a name of a UNIX shared object, and a name of a dynamically linked library module.

5. (original) The logical device as set forth in Claim 1 wherein said RTAP comprises a function selected from the group of a logical device, a device address, a name of a JAVA class, a name of a UNIX shared object, and a name of a dynamically linked library module.

6. (original) The logical device as set forth in Claim 1 wherein said RTAP and said directory attribute processor are configured to handle Lightweight Directory Access Protocol requests for attribute values.

7. (currently amended) The logical device as set forth in Claim 1 wherein said directory attribute processor is configured to disallow attribute modify requests for attributes declared as real-time ~~dynamic~~.

8. (currently amended) A method for dynamically handling ~~dynamic~~ real-time attributes in a static directory server comprising:

providing at least one declaration for an attribute to be dynamically handled as a ~~dynamic~~ real-time attribute in association with a set of directory attribute declarations, the value of said real-time attribute being retrievable outside of static memory of a directory structure;

parsing requests for access to directory attribute values to detect requests for attributes declared as ~~dynamic~~ real-time in said attribute declarations;

invoking at least one Real-time Attribute Processor (RTAP) selected according to a predetermined selection schema, said invoked RTAP being configured to dynamically resolve ~~determine~~ a ~~dynamic~~ real-time value for an attribute declared as being ~~dynamic~~ real-time in said set of attribute declarations by obtaining an attribute value from a real-time source external to said directory structure, and by converting said obtained value to conform to a directory request return format ~~, said dynamic value being unavailable from said static directory~~; and

returning to a requester an attribute value determined by said invoked RTAP.

9. (original) The method as set forth in Claim 8 wherein said step of selecting and invoking a RTAP selector comprises selecting an RTAP based upon a variation of a name of said requested directory attribute.

10. (original) The method as set forth in Claim 9 wherein said step of selecting an RTAP based upon an attribute name variation comprises selecting an RTAP from the group of a logical device, a device address, a name of a JAVA class, a name of a UNIX shared object, and a name of a dynamically linked library module.

11. (original) The method as set forth in Claim 8 wherein said step of invoking an RTAP comprises invoking an RTAP from the group of a logical device, a device address, a name of a JAVA class, a name of a UNIX shared object, and a name of a dynamically linked library module.

12. (original) The method as set forth in Claim 8 wherein said step of parsing a request comprises parsing a Lightweight Directory Access Protocol requests for attribute values.

13. (currently amended) The method as set forth in Claim 8 wherein said step of returning to a requester an attribute value ~~comprising~~ comprises returning said value according to a Lightweight Directory Access Protocol.

14. (currently amended) An article of manufacture ~~A computer-readable medium encoded with software for handling dynamic attributes in a static directory server, said software performing the steps~~ comprising:

a computer readable medium suitable for encoding software programs; and

one or more software programs encoded by said medium and configured to cause a processor to perform the steps of:

(a) providing at least one declaration for an attribute to be dynamically handled as a ~~dynamic~~ real-time attribute in association with a set of directory attribute declarations, the value of said real-time attribute being retrievable outside of static memory of a directory structure;

(b) parsing requests for access to directory attribute values to detect requests for attributes declared as ~~dynamic~~ real-time in said attribute declarations;

(c) invoking at least one Real-time Attribute Processor (RTAP) selected according to a predetermined selection schema, said invoked RTAP being configured to dynamically resolve ~~determine~~ a ~~dynamic~~ real-time value for an attribute declared as being ~~dynamic~~ real-time in said set of attribute declarations by obtaining an attribute value from a real-time source external to said directory structure, and by converting said obtained value to conform to a directory request return format ~~, said dynamic value being unavailable from said static directory~~; and

(d) returning to a requester an attribute value determined by said invoked RTAP.

15. (currently amended) The ~~medium~~ article as set forth in Claim 14 wherein said software for selecting and invoking an RTAP selector comprises software for selecting an RTAP based upon a variation of a name of said requested directory attribute.

16. (currently amended) The ~~medium~~ article as set forth in Claim 15 wherein said software for selecting an RTAP based upon an attribute name variation comprises software for selecting an RTAP from the group of a logical device, a device address, a name of a JAVA class, a name of a UNIX shared object, and a name of a dynamically linked library module.

17. (currently amended) The ~~medium~~ article as set forth in Claim 14 wherein said software for invoking an RTAP comprises software for invoking an RTAP from the group of a logical device, a device address, a name of a JAVA class, a name of a UNIX shared object, and a name of a dynamically linked library module.

18. (currently amended) The ~~medium~~ article as set forth in Claim 14 wherein said software for parsing a request comprises software for parsing a Lightweight Directory Access Protocol requests for attribute values.

19. (currently amended) The ~~medium~~ article The medium as set forth in Claim 14 wherein said software for returning to a requester an attribute value comprising software for returning said value according to a Lightweight Directory Access Protocol.